



# Réaliser un traitement dans un environnement Big Data sur le cloud



Soutenance Projet 9: Habibatou BA 05-04-2024

# Sommaire

- ❖ Problématique
- ❖ Les Données
- ❖ Processus de création de l'environnement Big Data
- ❖ Chaîne de traitement d'images
- ❖ Démonstration sur le cloud

# Problématique



Start-up **Fruits**  
**L'IA** au service  
de l'agriculture



**Etape 1: *application mobile grand public* de reconnaissance de fruit par photographie**

- Classification d'images (volume exponentiel d'images)



**Etape 2 :**  
***robots cueilleurs intelligents***

- Dans une mission ultérieure



## Mission :

- Mettre en place une architecture **Big Data**
- **Pré-Processing**
- **Réduction de dimension**

## Contraintes :

- Anticiper le passage à l'échelle (volume exponentiel, calculs distribués)
- Scripts en PySpark
- Déploiement Cloud



## Objectifs:

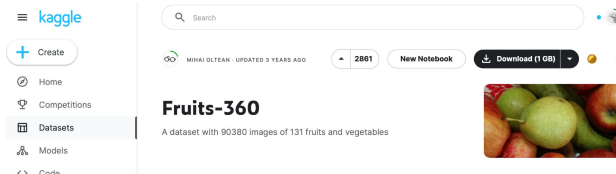
- Faire connaître la start-up
- Classification d'images pour application mobile

# Les Données

Image de 1 fruit ou 1 légume  
120 Variétés différentes



- ❖ Jeu de Test contenant **90380** images de fruits
- ❖ **131 Classes** ⇒ *Apple golden, Banana, Kiwi ...*
- ❖ Un répertoire par classe avec **plusieurs photos** du même fruit sous **différents angles**
- ❖ **Taille des images 100x100 pixels**
- ❖ Sur fond blanc uniformisé



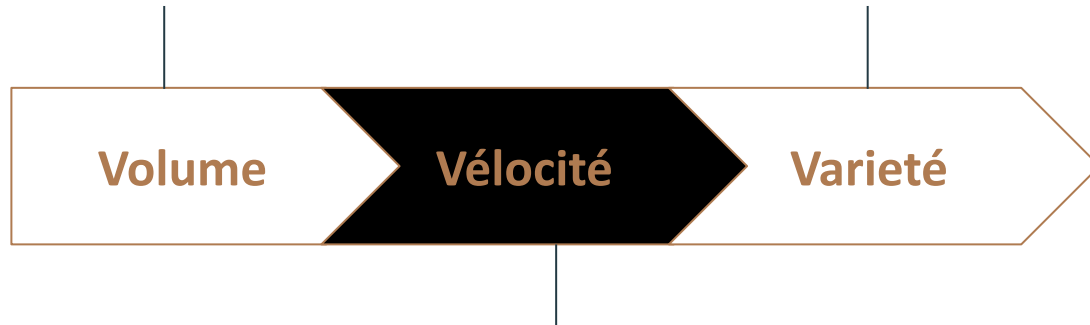


# Mise en œuvre du Processus de Big Data

# Pourquoi le BIG DATA ?

**Big Data (ou données massives)** : données telles que les solutions classiques de stockage, de gestion et de traitement ne suffisent plus.

- Avec une croissance rapide de l'application, le volume de données collectées augmentera considérablement.
- Le Big Data permet de stocker, gérer et analyser ces grandes quantités de données de manière évolutive.
- L'application mobile générera différents types de données, tels que des images de fruits, des informations associées et des métadonnées.
- Le Big Data permet de traiter et d'analyser efficacement ces multiples sources de données hétérogènes.



- Le nombre d'utilisateurs et de photos prises augmentera rapidement, générant une quantité importante de données en temps réel.
- Le Big Data offre les outils nécessaires pour traiter et analyser ces données à grande vitesse.



# Les outils



## Application Spark

Le **driver** distribue et planifie les tâches entre les différents **exécuteurs** qui les exécutent et permettent un traitement réparti. Il est le responsable de l'exécution du code sur les différentes machines.

**Cluster Manager** : assure le suivi des ressources disponibles.

- **Calculs distribués** : distribution du stockage et des traitement des données sur plusieurs unités de calcul réparties en clusters, au profit d'un seul projet afin de diviser le temps d'exécution d'une requête.
- **Apach Spark** : framework open-source permettant de traiter des bases de données massives en utilisant le calcul distribué (in-memory). Outil qui permet de gérer et de coordonner l'exécution de tâches sur des données à travers un groupe d'ordinateurs.
- **Algorithme MapReduce** :
  - Largement utilisé pour le traitement parallèle et distribué de grandes quantités de données.
  - Permet de diviser les données en ensembles plus petits, de les traiter indépendamment (MAP) et de les agréger pour obtenir le résultat final (REDUCE).
- Développement des scripts en **pySpark**, la librairie python (proche de pandas) permettant de communiquer avec Spark.  
⇒ **Avantages** : évolutivité (ajout de ressources supplémentaires), performances (accélération du temps de calculs), tolérance aux pannes (plus résilients aux pannes ou erreurs).

# Déploiement de la solution dans le cloud

- **Louer de la puissance de calcul à la demande** : pouvoir, quel que soit la charge de travail, obtenir suffisamment de puissance de calcul pour pouvoir traiter les images, même si le volume de données venait à fortement augmenter.
- **Diminuer les coûts** si l'on compare les coûts d'une location de serveur complet sur une durée fixe (1 mois, 1 année...).
- Le prestataire le plus connu et qui offre à ce jour l'offre la plus large dans le cloud est **Amazon Web Services (AWS)**.

The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and user information. Below this, the main content area is titled 'Page d'accueil de la console'. On the left, there's a sidebar with 'Récemment visité' (Recently visited) services like 'Gestion de la facturation et des coûts', 'EMR', 'S3', 'Elastic Beanstalk', 'EC2', 'IAM', and 'Elastic Container Service'. The main area is divided into two sections: 'Applications (0)' and 'Créer une application'. The 'Applications' section shows a dropdown for 'eu-west-3 (Région actuelle)' and a search bar. Below this, there's a table with columns 'Nom', 'Description', 'Région', and 'Compte d'origine', which is currently empty. A message states 'Aucune application' and encourages creating one. At the bottom, there's a button 'Créer une application' and a link 'Accédez à myApplications'.





# Configuration de l'environnement

The screenshot shows the AWS IAM console interface. The left sidebar contains navigation links: Identity and Access Management (IAM), Rechercheur sur IAM, Tableau de bord, Gestion des accès (highlighted), Groupes d'utilisateurs (highlighted), Utilisateurs, Rôles (highlighted), Politiques, Fournisseurs d'identité, Paramètres du compte, Rapports d'accès, Analyseur d'accès, Accès externe, Accès non utilisé, Paramètres de l'analyseur, Rapport sur les informations d'identification, and Activité de l'organisation. The main content area displays the details for a user named 'habiba'. The 'Informations d'identification de sécurité' dropdown menu is open, showing options: ID de compte: 9054-1832-1354, Compte, Organisation, Service Quotas, Gestion de la facturation et des coûts, and Informations d'identification de sécurité (highlighted). Below this, the 'Authentification multi-facteur (MFA)' section shows a warning: 'Aucun périphérique MFA. Attribuez un périphérique MFA pour améliorer la sécurité de votre environnement AWS.' The 'Clés d'accès (1)' section shows a table with one access key:

Identifiant de la clé d'accès	Créé le	Dernière utilisation de la clé d'accès	Dernière région utilisée	Dernier service utilisé
AKIA5FTZDFXFE4FTNV3	Il y a 1 heure	Il y a 6 minutes	eu-west-3	s3

- **Service IAM (Identity and Access Management)**
  - **Création d'un utilisateur**
  - **Gestion des droits (contrôle S3) (Politiques)**
  - **Création d'une paire de clés qui nous permettra de nous connecter de façon systématique login/mot de passe (*Informations d'identification de sécurité / Créer une clé d'accès*)**
- Installation et configuration de **AWS Cli** ( interface en ligne de commande d'AWS, permet d'interagir avec les différents services d'AWS)

# Stockage des données sur S3

[Amazon S3](#) > [Compartiments](#) > p9-habi-data

p9-habi-data [Info](#)

[Objets](#) | [Propriétés](#) | [Autorisations](#) | [Métriques](#) | [Gestion](#) | [Points d'accès](#)





Objets (4) [Info](#)

  Copier l'URI S3  Copier l'URL  Télécharger  Ouvrir  Supprimer  Actions ▼

[Créer un dossier](#) [Charger](#)

Les objets sont les entités fondamentales stockées dans Amazon S3. Vous pouvez utiliser l'[inventaire Amazon S3](#) pour obtenir une liste de tous les objets de votre compartiment. Pour que d'autres personnes puissent accéder à vos objets, vous devez leur accorder explicitement des autorisations. [En savoir plus](#)

 Rechercher des objets en fonction du

<input type="checkbox"/>	Nom ▲	Type ▼	Dernière modification ▼	Taille ▼	Classe de stockage ▼
<input type="checkbox"/>	 <a href="#">bootstrap-emr.sh</a>	sh	02 Apr 2024 07:52:35 PM CEST	339.0 o	Standard
<input type="checkbox"/>	 <a href="#">jupyter/</a>	Dossier	-	-	-
<input type="checkbox"/>	 <a href="#">Results/</a>	Dossier	-	-	-
<input type="checkbox"/>	 <a href="#">Test/</a>	Dossier	-	-	-

## S3 : Solution pour la gestion du stockage des données

- Stockage d'une grande variété d'objets (fichiers, image, vidéos...)
- Évolutivité avec espace disponible illimité.
- Indépendant des serveurs EC2.
- Accès aux données très rapide.
- Possibilité de définir des politiques d'accès IAM pour contrôler les autorisations. d'accès aux buckets et aux objets.
- Chiffrement côté serveur pour sécuriser les données stockées dans S3.
- Classes de stockage (options) adaptées à l'utilisation.

## Mise en oeuvre :

- Création d'un compartiment ("bucket") : **p9-habi-data**
- Choisir la même région pour les serveurs EC2 et S3.
- Chargement des données sur le bucket S3 :
  - Fichier de configuration avec amorçage
  - Répertoire des images **Test/**
  - Notebook avec Script (JupyterHub)
- Écriture des résultats dans le répertoire **Results**.

# EMR : Calculs Distribués



**Création du serveur EMR en 4 étapes :**

1. Configuration logiciel
2. Configuration matériel
3. Actions d'amorçage
4. Options de sécurité

**Elastic MapReduce (EMR)** : plateforme permettant d'exécuter des traitements de données distribuées à **grande échelle**, en utilisant des frameworks tels que Hadoop et Spark.

- Il utilise des **instances EC2** (Elastic compute cloud, serveur) avec des **applications préinstallées** et configurées pour créer et gérer le cluster de calculs distribués.

- Le service est **entièrement géré par AWS**.

⇒ **Avantages** : évolutivité, flexibilité, gestion simplifiée.

# EMR : Configuration logiciel



aws Services Rechercher [Alt+S] Francfort huba

Amazon EMR > EMR sur EC2: Clusters > Créer un cluster

## Créer un cluster

**Nom et applications - required**

clustP9

Version Amazon EMR: emr-7.0.0

Offre d'applications

Spark Interactive	Core Hadoop	Flink	HBase	Presto	Trino	Custom
-------------------	-------------	-------	-------	--------	-------	--------

☐ AmazonCloudWatchAgent 1.300051.1  
☐ Hue 4.11.0  
☐ Livy 0.7.1  
☐ Rhonin 5.1.1  
☒ Spark 3.5.0  
☐ Tez 0.10.2  
☐ ZooKeeper 3.5.10

☐ Flink 1.18.0  
☐ Hadoop 3.6  
☐ JupyterEnterpriseGateway 2.6.0  
☐ MINet 1.9.1  
☐ Pig 0.17.0  
☐ Sqoop 1.4.7  
☐ Trino 426

☐ HBase 2.4.17  
☐ Hive 3.1.3  
☐ JupyterHub 1.5.0  
☐ Oozie 3.2.1  
☐ Presto 0.285  
☒ TensorFlow 2.11.0  
☐ Zeppelin 0.10.1

Paramètres du catalogue de données AWS Glue

Options du système d'exploitation

Version Amazon Linux : ☒ Appliquez automatiquement les dernières mises à jour Amazon Linux

**Récapitulatif**

Nom et applications - required

clustP9

Version Amazon EMR: emr-7.0.0

Offre d'applications: Custom (Hadoop 3.6, JupyterHub 1.5.0, Spark 3.5.0, TensorFlow 2.11.0)

Configuration de cluster - required

Groupes d'instances uniformes

Primaire (m5.xlarge), Unité principale (m5.xlarge), Tâche (m5.xlarge)

Dimensionnement et mise en service du cluster - required

Configuration de mise en service

Taille du noyau: 1 Instance

Configuration des rôles IAM

Choisir un rôle IAM

Annuler Créer un cluster

## ■ Choix des logiciels :

- Hadoop et Spark : calculs distribués.
- TensorFlow : import du modèle et transfert learning.
- JupyterHub : exécution des scripts Pyspark du Notebook.

## ■ Paramétrage de la persistance des notebooks créés et ouverts via JupyterHub (configuration au format JSON).

### ▼ Paramètres du logiciel

Override the default configurations for specific applications on your cluster.

Entrer la configuration

Charger JSON à partir d'Amazon S3

```
1 {
2   "classification": "jupyter-s3-conf",
3   "properties": {
4     "s3.persistence.bucket": "p9-habi-data",
5     "s3.persistence.enabled": "true"
6   }
7 }
8
9 }
```

# EMR : Configuration Matériel



aws Services Rechercher [Alt+S]

Choisissez une méthode de configuration pour les groupes de nœuds primaires, principaux et de tâches de votre cluster.

☒ **Groupes d'instances uniformes**  
Choisissez le même type d'instance EC2 et la même option d'achat (à la demande ou Spot) pour tous les nœuds de votre groupe de nœuds. [En savoir plus](#)

☐ **Flottes d'instances flexibles**  
Choisissez parmi la plus grande variété d'options de provisionnement pour les instances EC2 de votre cluster. Diversifiez les types d'instances et les options d'achat, et utilisez une stratégie d'allocation. [En savoir plus](#)

### Groupes d'instances uniformes

#### Primaire

Choisir un type d'Instance EC2

**m5.xlarge**  
4 vCore 16 GiB mémoire EBS uniquement stockage  
Prix à la demande : 0.230 USD par instance/heure  
Prix Spot le plus bas : 0.071 USD (eu-central-1c)

Actions ▼

☐ **Utiliser la haute disponibilité**  
Lancez des clusters hautement disponibles et plus résilients avec trois nœuds primaires sur des instances à la demande. Cette configuration s'applique pendant toute la durée de vie de votre cluster. [En savoir plus](#)

► Configuration de nœud - facultatif

#### Unité principale

Choisir un type d'Instance EC2

**m5.xlarge**  
4 vCore 16 GiB mémoire EBS uniquement stockage  
Prix à la demande : 0.230 USD par instance/heure  
Prix Spot le plus bas : 0.071 USD (eu-central-1c)

Actions ▼

► Configuration de nœud - facultatif

## Configuration Matériel (choix des instances) :

- 1 instance **Maître** (driver), 2 instances **principales** (workers)
- Instances de **type M5** (instances de type équilibrées), et **xlarge** (la moins onéreuse disponible).



vCPU : 4 / Mémoire (Gio) : 16

Bande passante réseau (Gbit/s) : jusqu'à 10  
Bande passante EBS (Mbit/s) : Jusqu'à 4 750

# EMR : Bootstrapping

The screenshot shows the 'Ajouter une action d'amorçage' (Add bootstrap action) dialog box in the AWS EMR console. It includes fields for 'Nom' (Name) with the value 'ActionP9', 'Emplacement du script' (Script location) with the value 's3://p9-habi-data/bootstrap-emr.sh', and a section for 'Arguments - facultatif' (Optional arguments) with a text area for specifying arguments. At the bottom, there are 'Annuler' (Cancel) and 'Ajouter une action d'amorçage' (Add bootstrap action) buttons.

▼ Bootstrap actions (1) [Info](#) [Supprimer](#) [Modifier](#) [Ajouter](#)

Utilisez les actions d'amorçage pour installer des logiciels ou personnaliser la configuration de votre instance.

Nom	Emplacement Amazon S3	Arguments
<input type="radio"/> ActionP9	<a href="https://s3.amazonaws.com/p9-habi-data/bootstrap-emr.sh">s3://p9-habi-data/bootstrap-emr.sh</a>	-

- Choix des **packages manquants à installer**, utiles pour l'exécution du notebook.
- **A l'initialisation du serveur**, afin que les packages soient installés sur l'ensemble des machines du cluster et non pas uniquement sur le driver.
- Création du fichier "**bootstrap-emr.sh**" contenant commandes "**pip install**" pour installer les bibliothèques manquantes, et chargement sur le compartiment S3 (racine).
- Ajout du script dans les **actions d'amorçage**.

The screenshot shows a terminal window with the following commands and output:

```
#!/bin/bash
sudo python3 -m pip install -U setuptools
sudo python3 -m pip install -U pip
sudo python3 -m pip install wheel
sudo python3 -m pip install pillow
sudo python3 -m pip install pandas==1.2.5
sudo python3 -m pip install pyarrow
sudo python3 -m pip install boto3
sudo python3 -m pip install s3fs
sudo python3 -m pip install fsspec
```

The terminal status bar at the bottom shows the file path `~/Projet9/bootstrap-emr.sh`, line 10L, column 339B, and a status of 1,1 Tout.

# EMR : Sécurité

## ▼ Configuration de sécurité et paire de clés EC2 [Info](#)

Choose a security configuration or create a new one that you can reuse with other clusters.

### Configuration de sécurité

Sélectionnez les paramètres de chiffrement, d'authentification, d'autorisation et de service de métadonnées d'instance de votre cluster.

🔍 Choisir une configuration de sécurité



Parcourir

Créer une configuration de sécurité

### Paire de clés Amazon EC2 pour SSH sur le cluster [Info](#)

🔍 habi-cle-ec2



Parcourir

Créer une paire de clés

## ▼ Rôle Identity and Access Management (IAM) - *required* [Info](#)

Choisissez ou créez une fonction du service et un profil d'instance pour les instances EC2 de votre cluster.

### Fonction du service Amazon EMR [Info](#)

La fonction du service est un rôle IAM assumé par Amazon EMR pour mettre en service des ressources et effectuer des actions au niveau du service avec d'autres services AWS.

#### ☒ Choisir une fonction du service existant

Sélectionnez une fonction du service par défaut ou un rôle personnalisé avec des stratégies IAM attachées afin que votre cluster puisse interagir avec d'autres services AWS.

#### ☐ Créez une fonction du service

Laissez Amazon EMR créer une nouvelle fonction du service afin que vous puissiez accorder et restreindre l'accès aux ressources d'autres services AWS.

Fonction du service

EMR\_DefaultRole



### Profil d'instance EC2 pour Amazon EMR

Le profil d'instance attribue un rôle à chaque instance EC2 d'un cluster. Le profil d'instance doit spécifier un rôle qui peut accéder aux ressources pour vos étapes et actions d'amorçage.

#### ☒ Choisir un profil d'instance existant

Sélectionnez un rôle par défaut ou un profil d'instance personnalisé avec des stratégies IAM attachées afin que votre cluster puisse interagir avec vos ressources dans Amazon S3.

#### ☐ Choisir un profil d'instance

Laissez Amazon EMR créer un profil d'instance afin de pouvoir spécifier un ensemble personnalisé de ressources auquel il peut accéder dans Amazon S3.

Profil d'instance

EMR\_EC2\_DefaultRole



### Rôle d'autoscaling personnalisé - *facultatif*

Lorsqu'une règle d'autoscaling personnalisée se déclenche, Amazon EMR assume ce rôle pour ajouter et résilier les instances EC2. En savoir plus

Rôle d'autoscaling personnalisé

EMR\_AutoScaling\_DefaultRole



Créer un rôle IAM

- Sélection de la **paire de clés EC2** créée précédemment.
- Permet de se connecter en ssh aux instances EC2 sans avoir à entrer login / mot de passe.

⇒ **Création du cluster, instanciation du serveur** (statut *“En attente”*)

# EMR : Tunnel SSH => EC2

[illegible]

**Objectif : pouvoir accéder à nos applications en créant un tunnel SSH vers le driver.**

- Modification du **groupe de sécurité EC2** du driver :
  - Autorisation sur les connexions entrantes du driver : **ouverture du port 22** (port d'écoute du serveur SSH).

- Création du **tunnel SSH vers le driver** avec **Putty**.
- Configuration de **FoxyProxy**: redirection des requêtes vers le port 5555.
- Accès aux applications du serveur EMR via le tunnel SSH

⇒ Connexion au Notebook  
JupyterHub et exécution du code.





# Chaîne de traitement d'image

# Traitement des images

- Exécution du Notebook depuis **JupyterHub**, hébergé sur notre serveur EMR.
- Utilisation d'un **kernel pySpark**.
- Démarrage d'une **session Spark** à l'exécution de la première cellule.

## Chargement des données

- Images stockées dans un compartiment S3.
- Chargement des images dans des Spark DataFrame.

## Preprocessing

- Utilisation librairie PIL
- Redimensionnement des images  
(100,100,3)  $\Rightarrow$  (224, 224, 3)
- Fonction de preprocessing spécifique à MobileNet

## Extraction de features - Réduction de dimensions

- Modèle MobileNetV2, pré-entraîné sur la base imageNet.
- Couche de sortie : avant dernière couche (extraction de features)
- Extraction de features par batch à l'aide de pandas UDF
- Réduction PCA

## Stockage des résultats

- Écriture des résultats dans des fichiers Parquet.
- Stockage dans le compartiment S3.

# Chargement des images

Entrée [4]: `images.show(10)`

path	modificationTime	length	content
s3://p9-habi-data...	2024-03-30 15:37:00	125135	[FF D8 FF E0 00 1...]
s3://p9-habi-data...	2024-03-30 15:37:00	124785	[FF D8 FF E0 00 1...]
s3://p9-habi-data...	2024-03-30 15:37:00	123514	[FF D8 FF E0 00 1...]
s3://p9-habi-data...	2024-03-30 15:37:00	122958	[FF D8 FF E0 00 1...]
s3://p9-habi-data...	2024-03-30 15:37:00	122807	[FF D8 FF E0 00 1...]
s3://p9-habi-data...	2024-03-30 15:37:00	122654	[FF D8 FF E0 00 1...]
s3://p9-habi-data...	2024-03-30 15:37:00	122470	[FF D8 FF E0 00 1...]
s3://p9-habi-data...	2024-03-30 15:37:01	121883	[FF D8 FF E0 00 1...]
s3://p9-habi-data...	2024-03-30 15:37:00	121883	[FF D8 FF E0 00 1...]
s3://p9-habi-data...	2024-03-30 15:37:00	121530	[FF D8 FF E0 00 1...]

```
root
|-- path: string (nullable = true)
|-- modificationTime: timestamp (nullable = true)
|-- length: long (nullable = true)
|-- content: binary (nullable = true)
|-- label: string (nullable = true)
```

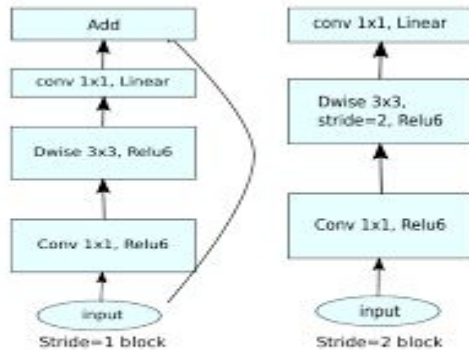
None

path	label
s3://p9-habi-data/Test/apple_hit_1/r0_115.jpg	apple_hit_1
s3://p9-habi-data/Test/apple_hit_1/r0_119.jpg	apple_hit_1
s3://p9-habi-data/Test/apple_hit_1/r0_107.jpg	apple_hit_1
s3://p9-habi-data/Test/apple_hit_1/r0_143.jpg	apple_hit_1
s3://p9-habi-data/Test/apple_hit_1/r0_111.jpg	apple_hit_1
s3://p9-habi-data/Test/apple_hit_1/r0_127.jpg	apple_hit_1
s3://p9-habi-data/Test/apple_hit_1/r0_139.jpg	apple_hit_1
s3://p9-habi-data/Test/apple_hit_1/r0_123.jpg	apple_hit_1
s3://p9-habi-data/Test/apple_hit_1/r0_151.jpg	apple_hit_1
s3://p9-habi-data/Test/apple_hit_1/r0_103.jpg	apple_hit_1

only showing top 10 rows

- Chargement des données avec `spark.read()` :
  - Traitement des fichiers en tant que **données binaires**.
  - À l'emplacement spécifié (compartiment S3), recherche récursive dans les sous-répertoires des fichiers avec l'extension **".jpg"**.
  - Chargement des images dans un **DataFrame Spark**.
- Ajout de la colonne **label** issu du chemin d'accès des fichier :
  - **label** représente la catégorie de l'image (nom du fruit), avant dernier élément (-2) du "path".

# MobilNetV2 avec TL



(d) Mobilenet V2

- ❖ **MobileNetV2** : CNN pré-entraîné sur la base ImageNet pour la détection de features et la classification d'images.
- ❖ **Transfer Learning** : Création une instance du modèle MobileNetV2 pré-entraîné avec les poids du jeu de données ImageNet, incluant la couche de classification finale.
- ❖ **Préparation du modèle** :
  - Création d'un nouveau modèle avec pour couche de sortie l'avant-dernière couche (extraction des features images) du modèle MobileNetV2 .
  - Dimension vecteur de sortie (1, 1, 1280).
  - Diffusion des poids avec `sparkContext.broadcast()` de PySpark :
    - ○ Chargement du modèle sur le driver puis diffusion des poids aux workers.
    - ○ Permet de distribuer une variable à travers le cluster afin qu'elle soit disponible pour tous les nœuds de calcul.

# PreProcessing



- Dimensions des **images d'origine** : **(100,100,3)** / (100\*100 pixels et 3 canaux de couleur RVB).
- Dimensions des **images attendues en entrée de MobileNetV2** : **(224, 224, 3)**  
⇒ **Nous devons les redimensionner avant de les confier en entrée du modèle.**
- Avec **librairie PIL** (Python Imaging Library) :
  - Ouverture des données binaires de l'image en tant qu'image.
  - **Redimensionnement de l'image** à une taille (224, 224,3).
- Application de la fonction *preprocess\_input* de TensorFlow, **fonction de prétraitement spécifique** pour prétraiter les images avant de les passer en entrée du modèle MobileNet.

# Traitements

Amazon S3

Amazon S3 > Compartiments > p9-habi-data > Results/

Results/

Objets Propriétés

Objets (26) Info

Créer un dossier Charger


Les objets sont les entités fondamentales stockées dans Amazon S3. Vous pouvez utiliser l'[Inventaire Amazon S3](#) pour obtenir une liste de tous les objets de votre compartiment. Pour que d'autres personnes puissent accéder à vos objets, vous devez leur accorder explicitement des autorisations. [En savoir plus](#)

Rechercher des objets en fonction de

<input type="checkbox"/>	Nom	Type	Dernière modification	Taille	Classe de stockage
<input type="checkbox"/>	_SUCCESS	-	02 Apr 2024 08:14:55 PM CEST	0 o	Standard
<input type="checkbox"/>	part-00000-9964e765-b1d6-4488-a856-2a13f1da1ef4-c000.snappy.parquet	parquet	02 Apr 2024 08:14:23 PM CEST	403.1 Ko	Standard
<input type="checkbox"/>	part-00001-9964e765-b1d6-4488-a856-2a13f1da1ef4-c000.snappy.parquet	parquet	02 Apr 2024 08:14:23 PM CEST	411.7 Ko	Standard

- **Extractions des Features**
- Réduction de dimension **PCA** : **Analyse en composantes principales** pour réduire la dimensionnalité tout en préservant un maximum d'informations.
- **Stockage des résultats** :
  - Données du DataFrame écrites dans un fichier Parquet (format de stockage optimisé pour le Big Data).
  - Mode "overwrite" : si le fichier de destination existe déjà, il sera écrasé.
  - Dans le répertoire "Results" du compartiment S3.

# Démonstration d'exécution dans le cloud

 Ba\_Habibatou\_1bis\_notebookcloud\_032024 (modifié) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help

Non fiable | PySpark

```
19, 0.1504102490292923, 1.3289066553115845, 0.0, 0.369780957698822, 0.0, 0.10885540395975113, 0.0, 0.0, 0.272288948
2975806, 0.05801090970635414, 0.0, 0.0, 0.4923121929168701, 0.11733233183622236, 1.0896481275558472, 0.5527678132857
19, 0.13772350549697876, 0.0, 0.2897004187187086, 0.0, 0.0012587513774633408, 0.0, 0.0, 0.39889124035835266, 2.1622
540590775146, 0.10844777456521988, 0.7929095929838933, 0.0, 0.14325371384620667, 0.0, 0.4165956676066317, 0.0, 0.43
1431298702955246, 0.0, 0.08237455048216446, 1.3896169662475586, 0.2929629981517792, 0.0, 0.01572241075336933, 0.185
81536412239075, 0.22676408290863037, 0.0, 0.2588525550635448, 0.30766692757606506, 0.0, 0.14075791835784912, 0.1085
7957601547241, 0.07713747769594193, 2.2778561115264893, 0.0, 0.0, 0.032548874616622925, 0.0, 0.030957309529185295,
0.0, 0.15892843578624725, 0.8906516432762146, 0.0, 0.0, 0.01713264174759388, 0.002285152763853775, 0.26536127924919
13, 0.05566941574215809, 0.0, 0.05892209708690643, 0.0, 2.4857006637854434, 0.078535562324913, 0.312523424653967
3, 0.41594597697257996, 1.10013473083306503, 0.006748804822564125, 0.0, 0.5849771499633789, 0.0, 0.0, 0.1, 1.4742290
32325446, 0.9739450812239783, 0.31168297989845276, 1.523823618888855, 2.1509337425231934, 0.0, 0.0, 0.120696422161
57913, 0.5334492325782776, 0.7433648109436035, 0.0, 0.0, 0.05431675910949787, 0.0, 0.0, 0.007384500931948423, 0.140
29723405838013, 0.0, 0.049000807106494904, 0.0, 1.0548295974731445, 0.005689573474228382, 0.2784684308426685, 0.10
562962293624878, 0.301680803299502, 0.008641830645501614, 1.1209160089492798, 0.0, 0.24973316490050177, 0.0, 0.0,
```

Entrée [33]: df\_pca

```
DataFrame[path: string, label: string, features: array<float>, features_vectors: vector, features_scaled: vector, f
eatures_pca: vector]
```

Entrée [34]: print(df\_pca.printSchema())

```
root
|-- path: string (nullable = true)
|-- label: string (nullable = true)
|-- features: array (nullable = true)
|   |-- element: float (containsNull = true)
|   |-- features_vectors: vector (nullable = true)
|   |-- features_scaled: vector (nullable = true)
|   |-- features_pca: vector (nullable = true)
None
```

 Ba\_Habibatou\_1bis\_notebookcloud\_032024 (modifié) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help

Non fiable | PySpark

## 10 Exécution du code

Je décide d'exécuter cette partie du code depuis **JupyterHub hébergé sur notre cluster EMR**.  
Pour ne pas alourdir inutilement les explications du **notebook**, je ne réexpliquerai pas les étapes communes que nous avons déjà vues dans la première partie où l'on a exécuté le code localement sur notre machine virtuelle Ubuntu.

Avant de commencer, il faut s'assurer d'utiliser le **kernel pyspark**.

En utilisant ce kernel, une session spark est créé à l'exécution de la première cellule.  
Il n'est donc **plus nécessaire d'exécuter le code "spark = (SparkSession ...)"** comme lors de l'exécution de notre notebook en local sur notre VM Ubuntu.

### 10.1 Démarrage de la session Spark

Entrée [1]: *# L'exécution de cette cellule démarre l'application Spark*

Starting Spark application

ID	YARN Application ID	Kind	State	Spark UI	Driver log	Current session?
0	application_1712081113018_0001	pyspark	idle	<a href="#">Link</a>	<a href="#">Link</a>	✓

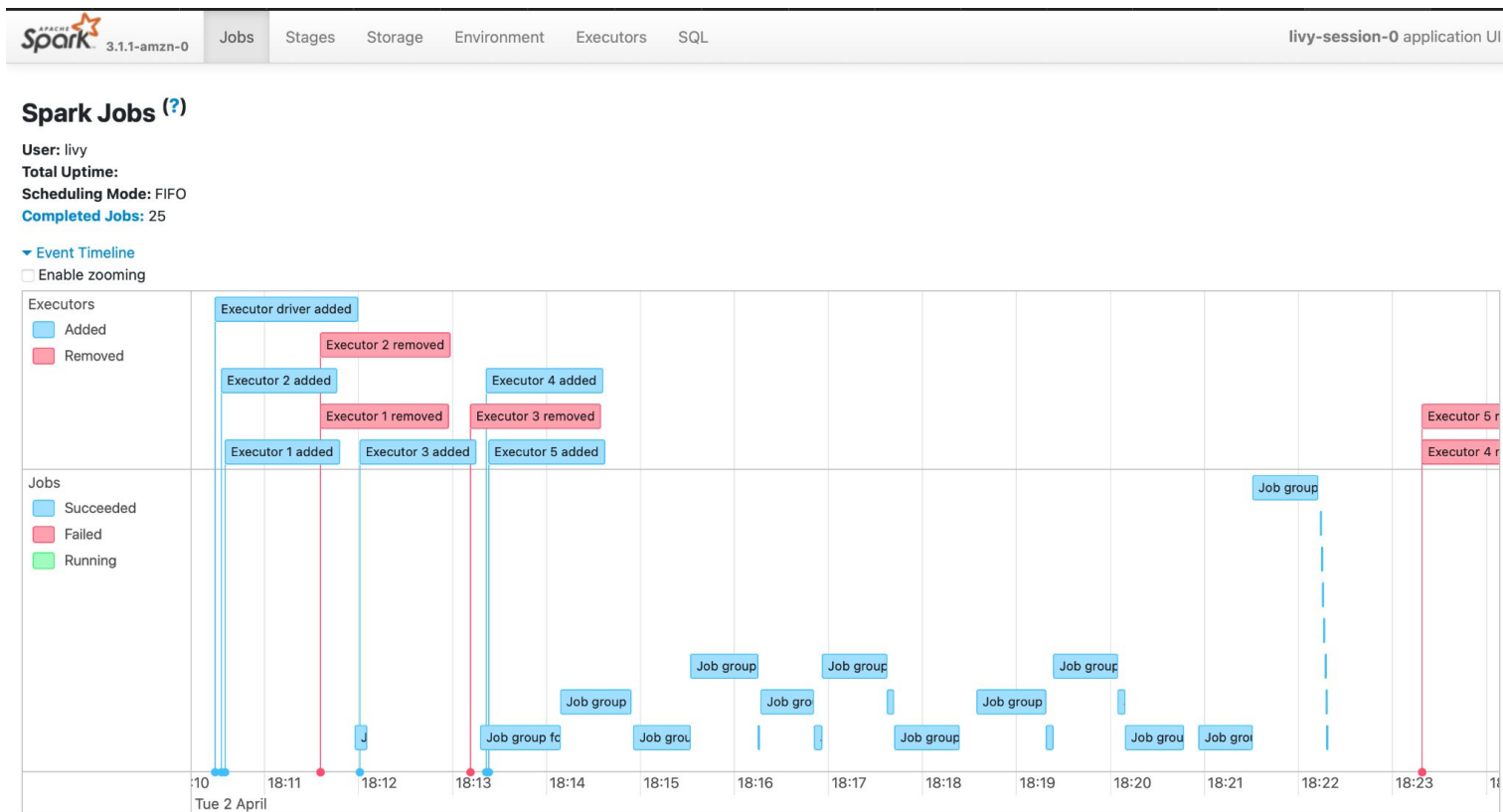
SparkSession available as 'spark'.

Entrée [2]: *#nettoyer la sortie standard*  

```
sc = spark.sparkContext
sc.setLogLevel('ERROR')
```

Affichage des informations sur la session en cours et liens vers Spark UI :

# Démonstration d'exécution dans le cloud





# Conclusion & Suite du Projet



- **Mise en place d'une architecture Big Data :**
  - **EMR** (Elastic MapReduce) avec Apache Spark pour le traitement distribué des données volumineuses, qui nous permet d'instancier un cluster avec les programmes et bibliothèques nécessaires : Spark, Hadoop, JupyterHub, TensorFlow...
  - **S3** (Simple Storage Service) pour le stockage des données : images d'origine et résultats.
  - **IAM** (Identity & Access Management) pour la gestion des contrôles d'accès.
- Appropriation de la **chaîne de traitement d'images** : chargement des données, preprocessing, préparation du modèle MobileNetV2 avec transfert learning et diffusion des poids, extraction de features, réduction de dimensions.

- **L'utilisation d'un environnement Big Data offre des avantages pour "Fruits!" en termes de traitement des données, de performance, d'évolutivité et de préparation pour l'avenir :**
  - Il sera facile de faire face à une montée de la charge de travail et **passer à l'échelle** en redimensionnant le cluster de machines.
  - Les coûts augmenteront en conséquence mais resteront inférieurs aux coûts engendrés par l'achat de matériels ou par la location de serveurs dédiés.
  - L'architecture Big Data jette les bases pour des fonctionnalités avancées, comme l'**entraînement de modèles de classification** des fruits.